Open-Source Technology Use Report

Proof of knowing your stuff in CSE312

# Guidelines

Provided below is a template you must use to write your report for each of the technologies you use in your project.

Here are some things to note when working on your report, specifically about the **General Information & Licensing** section for each technology.

* **Code Repository**: Please link the code and not the documentation. If you’d like to refer to the documentation in the **Magic** section, you’re more than welcome to, but we’d like to see the code you’re referring to as well.
* **License Type**: Three letter acronym is fine.
* **License Description**: No need for the entire license here, just what separates it from the rest.
* **License Restrictions**: What can you *not* do as a result of using this technology in your project? Some licenses prevent you from using the project for commercial use, for example.
* **Who worked with this?**: It’s not necessary for the entire team to work with every technology used, but we’d like to know who worked with what.

Also, feel free to extend the cell of any section if you feel you need more room.

If there’s anything we can clarify, please don’t hesitate to reach out! You can reach us using the methods outlined on the course website or see us during our office hours.

# Flask

## General Information & Licensing

|  |  |
| --- | --- |
| Code Repository | https://github.com/pallets/flask |
| License Type | BSD-3 |
| License Description | Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:   * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. * Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission. |
| License Restrictions | * Prohibits others from using the name of the copyright holder or its contributors to promote derived products without written consent. |
| Who worked with this? | The whole team, obviously – but so far, mostly Cole has implemented it. |

*Use as many of the sections below as needed, or create more, to explain every function, method, class, or object type you used from this library/framework.*

## Purpose

|  |
| --- |
| Replace this text with some that answers the following questions for the above tech:   * What does this tech do for you in your project?   + This is mostly for the HTTP handling so far. * Where specifically is this tech used in your project? Give us some details like file location and line number, if applicable. If too cumbersome, a general description of where it’s used for a given purpose is fine as well.   + It’s used in the “server.py” file. It’s all of the handles for every link related to the webpage like “webpage.com/here” where “here” is subject, Flask allows us to set what each of those redirects to – what it should serve the user.   + Also, it helps with GET, POST, etc. requests since it allows for separate handling based on what type of request – like in the homeworks but less masochistic. * Specific Lines and what they do   + @app.route(“/”) – (currently) line 11 in server.py     - This is used as a handler for if the user requests (GETs) the homepage (as “/” represents the homepage)     - The “.route” method is a decorator function of Flask used to make requesting more visually appealing (and more understandable to the programmers/developers)     - It uses the “add\_url\_rule” method which creates a url rule (kind of like a dictionary entry [at least I like to imagine so] where it’s {“url” -> do\_this} ) which creates a “if the user requests this path, they’ll always be lead here” in the app.   + render\_template(“index.html”) – (currently) line 13 in server.py     - This is the Flask method used to show the html to the user (to return it upon the user requesting (GETting) the “/” directory     - The render\_template method is used by Flask to send the HTML to the user’s browser.     - In the render\_template method, it first gets the current app (in this case app since that’s what we named our Flask(\_\_name\_\_)) and then it uses a get\_or\_select\_template from jinja\_env which will return a template (depending on the parameters passed – in our case it will be a regular “template\_name\_or\_list” – i.e. the index.html we passed it originally in the flask render\_template method) which will then use Flask’s method “\_render” which takes the app, the template, and the context (which there isn’t in our case). The “\_render” then checks the context (again, none), and creates a template before it’s rendered, then renders the template (which won’t make a difference since we aren’t using context – so the rendered and the unrendered version are going to be the same) and then it returns the rendered version of the template. This will be a string, it seems.   + It doesn’t say anywhere how it turns the HTML into an actual “this is what is going to be sent to the user” |

## *Magic* ⋆★͎۪۫｡˚۰˚☽˚⁀➷｡˚★彡͎۪۫⋆ ༄

|  |
| --- |
| Dispel the magic of this technology. Replace this text with some that answers the following questions for the above tech:   * How does this technology do what it does for you in the **Purpose** section of this report? Please explain this in detail, starting from after the TCP socket is created. Remember, to be allowed to use a technology in your project, you must be able to know how it works. * Where is the specific code that does what you use the tech for? You ***must*** provide a link to the specific file in the repository for your tech with a line number or number range.   + https://github.com/pallets/flask/blob/3dc6db9d0cfddcfb971c382b014bb56ac3761d3c/src/flask/scaffold.py#L423   + If there is more than one step in the chain of calls *(hint: there will be)*, you must provide links for the entire chain of calls from your code, to the library code that actually accomplishes the task for you.   + Example: If you use an object of type HttpRequest in your code which contains the headers of the request, you must show exactly how that object parsed the original headers from the TCP socket. This will often involve tracing through multiple libraries and you must show the entire trace through all these libraries with links to all the involved code.   \*This section may grow beyond the page for many features. |
|  |